

REV. 12/99  
For Other Than A Small Entity

Attorney Docket No. CURL-006

Applicants : Robert H. Halstead et al.  
For : INTERACTIVE EXAMPLES FOR ONLINE CODING  
TUTORIALS

EXPRESS MAIL CERTIFICATION

"Express Mail" mailing label number EL569205665US.

Date of Deposit September 28, 2000.

I hereby certify that this transmittal letter and the other papers and fees identified in this transmittal letter as being transmitted herewith are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above and are addressed to the Hon. Assistant Commissioner for Patents, Washington, D.C. 20231.



Lily Jiang

Hon. Assistant Commissioner  
for Patents  
BOX PATENT APPLICATION  
Washington, D.C. 20231

TRANSMITTAL LETTER FOR UNEXECUTED  
ORIGINAL PATENT APPLICATION

Sir:

Transmitted herewith for filing are the  
[X] specification; [X] claims; [X] abstract; [X] unexecuted  
declaration; [X] printEFS data sheet, for the above-identified  
patent application.

Also transmitted herewith are:

[X] 7 sheets of:

[ ] Formal drawings.

[X] Informal drawings. Formal drawings will be filed  
during the pendency of this application.

[ ] Certified copy(ies) of application(s)

(country)	(appln. no.)	(filed)
-----------	--------------	---------

(country)	(appln. no.)	(filed)
-----------	--------------	---------

(country)	(appln. no.)	(filed)
-----------	--------------	---------

from which priority is claimed.

[ ] An assignment of the invention to \_\_\_\_\_

[ ] A check in the amount of \$40.00 to cover the recording fee.

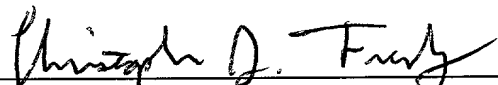
[ ] Please charge \$40.00 to Deposit Account No. 06-1075 in payment of the recording fee. A duplicate copy of this transmittal letter is transmitted herewith.

[ ] An associate power of attorney.

The filing fee has been calculated as shown below:

FOR	NUMBER FILED	NUMBER EXTRA	RATE	FEE
BASIC FEE				\$690.00
TOTAL CLAIMS	19 - 20 =	0	X \$18 =	\$0.00
INDEPENDENT CLAIMS	2 - 3 =	0	X \$78 =	\$0.00
[ ] MULTIPLE DEPENDENT CLAIMS			+ \$260 =	\$
			TOTAL	<u>\$690.00</u>

- [X] A check in the amount of \$ 690.00 in payment of the filing fee is transmitted herewith.
- [ ] This application is being filed unaccompanied by a filing fee. The appropriate filing fee will be paid in response to a Notice to File Missing Parts, pursuant to 37 C.F.R. § 1.53(d).
- [X] The Commissioner is hereby authorized to charge payment of any additional filing fees required under 37 C.F.R. § 1.16, in connection with the paper(s) transmitted herewith, or credit any overpayment of same, to deposit Account No. 06-1075. A duplicate copy of this transmittal letter is transmitted herewith.
- [ ] Amend the specification by inserting before the first line the sentence: -- This is a [ ] continuation-in-part, of application Serial No.: \_\_\_\_\_, filed \_\_\_\_\_  
entitled \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_.
- [ ] Please charge \$ \_\_\_\_\_ to Deposit Account No. 06-1075 in payment of the filing fee. A duplicate copy of this transmittal letter is transmitted herewith.

  
Christopher J. Frerking  
Registration No. 42,557  
Agent for Applicants

Nicola A. Pisano  
Registration No. 34,408  
Attorney for Applicants

FISH & NEAVE  
Customer No. 1473  
1251 Avenue of the Americas  
New York, New York 10020-1104  
Tel.: (650) 617-4000

## INVENTOR INFORMATION

Inventor One Given Name:: Robert H  
Family Name:: Halstead  
Postal Address Line One:: 24 Louise Road  
City:: Belmont  
State or Province:: Massachusetts  
Country:: US  
Postal or Zip Code:: 02178  
City of Residence:: Belmont  
State or Province of Residence:: Massachusetts  
Country of Residence:: US  
Citizenship Country:: US  
Inventor Two Given Name:: Patrick J  
Family Name:: LoPresti  
Postal Address Line One:: 61A 6th Street  
City:: Cambridge  
State or Province:: Massachusetts  
Country:: US  
Postal or Zip Code:: 02141  
City of Residence:: Cambridge  
State or Province of Residence:: Massachusetts  
Country of Residence:: US  
Citizenship Country:: US

## CORRESPONDENCE INFORMATION

Correspondence Customer Number:: 1473  
Fax One:: 212.596.9090

## APPLICATION INFORMATION

Title Line One:: INTERACTIVE EXAMPLES FOR ONLINE CODING T  
Title Line Two:: UTORIALS  
Total Drawing Sheets:: 7  
Formal Drawings?:: No  
Application Type:: Utility  
Docket Number:: CURL-006  
Secrecy Order in Parent Appl.?:: No

## REPRESENTATIVE INFORMATION

Representative Customer Number:: 1473  
Registration Number One:: 34408  
Registration Number Two:: 39164  
Registration Number Three:: 42557

Source:: PrintEFS Version 1.0.1

CURL-006

## INTERACTIVE EXAMPLES FOR ONLINE CODING TUTORIALS

Field Of The Invention

This invention relates generally to methods and apparatus for providing examples in online coding tutorials. More specifically, the present invention provides methods and apparatus for generating interactive coding examples in online tutorials for computer languages.

Background Of The Invention

The Internet and the World Wide Web (hereinafter "the web") have revolutionized the ways in which information is disseminated and shared. A wide variety of information can be simultaneously accessed by multiple users through a new category of documents designed to easily represent content for display and transmission over the Internet. These new documents, often referred to as *electronic documents* or *web pages*, are increasingly replacing their traditional paper counterparts as the medium through which business is carried out.

An electronic document is a multimedia composition that is displayed to the user on a "web browser window" by "web browser software". Electronic documents may contain text, audio, graphics, imagery, and video content, as well as nearly any other type of

content that may be experienced through use of a computer or other electronic devices. Additionally, electronic documents may be interactive, and may contain user selectable links that cause other  
5 electronic documents to be accessed.

Electronic documents are used in many creative and diverse fields, including entertainment, commerce, government, and education. In particular, the easy access to information in electronic documents  
10 has transformed education, enabling students and teachers to participate in a rich learning environment. Lectures, research materials, and other course components may be made available on demand, allowing students with a computer connected to the Internet to  
15 have access to educational material from any place at any time.

The Internet provides a very effective environment for the collaborative learning of various topics, especially when it involves the use of a  
20 computer, such as, for example, learning computer languages. Students can access a number of *online tutorials* on computer languages, learn about the languages' keywords and syntax, and immediately apply the concepts learned in the tutorials by writing and  
25 executing programs on the same computer where the tutorials are being displayed. The students can further share their programs with others for review, by including them in electronic documents accessible over the Internet. This process has considerably  
30 facilitated the learning of computer languages for many users, including the growing community of non-technically savvy users who desire to be proficient in creating computer applications.

Online computer language tutorials are

essentially a set of electronic documents explaining the language syntax and providing examples on how the language is used to implement computer programs. The examples are usually a set of instructions or exercises  
5 for the user to perform. Some online tutorials may provide an interactive window in the electronic document, such as an editing window, for the user to type in the example instructions. This activity takes place in the user's computer, with the web browser  
10 software displaying the example instructions and the interactive window. The user may select a "submit" button associated with the interactive window for execution of the example instructions. Once the submit button is selected, the web browser software  
15 establishes a connection over the Internet between the user's computer and a "web server", which is responsible for the execution of the instructions typed in the interactive window displayed in the user's computer. The example instructions are then processed  
20 by the web server, and the results are sent back to the user's computer over the Internet. The results of the example instructions are displayed in another electronic document in the user's computer once the web browser software handles the web server's response.

25           This type of online tutorial has several significant drawbacks. First, depending on the Internet connection the user has, it may take several minutes for the user to type the example instructions in the interactive window of the electronic document,  
30 send the example instructions for processing at the web server, and get the results back. Under heavy Internet congestion, it may take several processing requests before the web server responds with the example results. This problem is accentuated if the example

itself requires user interaction.

Second, this type of online tutorial does not enable the user to easily and quickly modify the example instructions to see how they could be adapted to a particular situation. Modifying the example instructions would require the user to perform several steps, including: (1) executing the original example instructions; (2) sending the appropriate requests to the server; (3) analyzing the example results; (4) generating the modified set of instructions; (5) sending the new requests to the server for executing the modified instructions; and (6) analyzing the new example results and comparing them to the original example results. This process may be repeated multiple times, and in certain circumstances, it may be hard for the user to keep track of which changes in the example instructions produced which set of results. This usually occurs when the results are displayed in several "pop-up windows" that may be accidentally closed by the user, or when the results are displayed in the same web browser window used by the tutorial to display the original instructions, which would require the user to select the "back" button on the web browser repeatedly to view the original instructions. The latter approach is the one adopted by an online tutorial on the SQL database language.

Finally, if the user makes an error when typing the example instructions, the server typically responds with an error message that simply says an error occurred without giving the user further indications of why the typed example instructions could not be executed properly. Users may get frustrated with the lack of guidance provided by such examples.

In view of the drawbacks above, it would be



desirable to provide methods and apparatus for generating coding examples in online tutorials for computer languages that overcome these drawbacks.

It would further be desirable to provide  
5 methods and apparatus for generating coding examples in electronic documents that can be processed in the user's computer.

It would still further be desirable to provide methods and apparatus for generating  
10 interactive coding examples in electronic documents that enable the user to easily view, modify, and execute the example instructions.

It would also be desirable to provide methods and apparatus for generating coding examples that  
15 provide suitable error messages when the user makes a mistake when writing the example source code.

#### Summary Of The Invention

In view of the foregoing, it is an object of the present invention to provide methods and apparatus  
20 for providing interactive coding examples in online computer language tutorials that overcome the various drawbacks described hereinabove.

It is another object of the present invention to provide methods and apparatus for generating coding  
25 examples in electronic documents that can be processed in the user's computer.

It is a further object of the present invention to provide methods and apparatus for generating interactive coding examples in electronic  
30 documents that enable the user to easily view, modify, and execute the example instructions.

It is also an object of the present invention to provide methods and apparatus for generating coding

examples that provide suitable error messages when the user makes a mistake when writing the example source code.

These and other objects of the present invention are accomplished by providing methods and apparatus for generating interactive coding examples that enable an user to easily view, modify, and execute the examples on the user's computer. The interactive examples are embedded in electronic documents that are part of online tutorials for computer languages.

In a preferred embodiment, the methods of the present invention for providing interactive examples in online coding tutorials involve three major steps: (1) defining an example markup in the language used to generate the electronic documents for the online tutorials; (2) processing the example content to convert it into a displayed form; and (3) creating an interactive example window.

More specifically, the present invention provides a specially designed example markup to create and display interactive example windows in online computer language tutorials. The interactive example window contains three sub-windows, hereinafter referred to as "widgets", consisting of: (1) an interactive editing widget for displaying the example content and enabling the user to modify the example content; (2) a control affordances widget containing a set of control buttons for the user to apply to the example content; and (3) an example result widget for displaying the example results and allowing the user to interact with the example result.

An important advantage of the present invention inheres in the fact that all operations performed by the user on the interactive example may be

processed in the user's computer. This avoids connecting to a web server, sending requests to the web server to perform the operations, and waiting for the web server to respond with the operations' results.

5           Another technical advantage of the present invention inheres in the fact that it enables users to easily and quickly modify the example source code, analyze the example results, debug the example content, and repeat the process using the same interactive  
10 example window.

Other technical advantages of the present invention will become apparent to one of ordinary skill in the art in view of the drawings and detailed description of the embodiments of the invention.

15 Brief Description Of The Drawings

The above and other objects of the present invention will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which like reference  
20 characters refer to like parts throughout, and in which:

FIG. 1 is a schematic view of a computer system suitable for use with the present invention;

FIG. 2 shows the network environment in which  
25 the present invention operates;

FIG. 3 illustrates the example markup used to include examples in online tutorials;

FIG. 4 is a flow chart for executing the example markup;

30           FIG. 5 is a schematic view of a preferred embodiment of an interactive example window for use with the present invention;

FIG. 6 is a schematic view of a preferred

embodiment of a control affordances widget for the interactive example window;

FIG. 7 shows an error message window for use with the present invention;

5           FIGS. 8A, 8B, 8C, 8D, and 8E show an example use of the present invention; and

          FIGS. 9A and 9B show an example web browser window displaying an online coding tutorial containing an interactive example window and the source code used  
10 to display the online coding tutorial and example in the web browser window.

#### Detailed Description Of The Invention

          The present invention provides interactive coding examples in online computer language tutorials  
15 that are designed as electronic documents. It should be noted that as used herein, an "electronic document" comprises a source file or collection of files written using one or more language tools for displaying content accessible with web browser software. The content may  
20 consist of text, audio, graphics, imagery, and video, as well as any other type of content that may be experienced through use of web browser software. Additionally, electronic documents may be interactive, and may contain user selectable links that cause other  
25 electronic documents to be accessed.

          As used herein, online computer language tutorials consist of a collection of electronic documents containing specialized content describing the computer language syntax and other language features  
30 and how they are used to implement computer programs. The computer language may be any type of language designed to enable humans to communicate with or control computers, including programming languages such

as C, markup languages such as HTML, script languages such as PERL, and database query languages such as SQL, among others. The online tutorials may contain coding examples showing how to use language elements and  
5 expressions to implement certain features in computer programs. The coding examples may also be interactive, containing control buttons for the user to operate to better understand how the coding example works.

Referring to FIG. 1, a computer system  
10 suitable for use with the present invention is described. Computer system 10 includes at least processor 11, for processing information according to programmed instructions, memory 12, for storing information and instructions for processor 11, storage  
15 system 13, such as a magnetic or optical disk system, for storing large amounts of information and instructions on a relatively long-term basis, and display system 14, such as a computer monitor, for displaying various graphical elements that facilitate  
20 user interaction with computer system 10. These graphical elements include graphical user interfaces (GUI) of software applications, windows for displaying Internet content through a web browser, and various "pop-up" windows that are triggered by user interaction  
25 with a given application, such as pop-up windows that display the results of an online coding example.

Memory 12 contains instructions for web browser software as well as for an "example execution engine", consisting of instructions for executing  
30 source code examples in online computer language tutorials. The execution engine may be embedded in a web browser, may be a software plug-in for the web browser, or it may be a stand-alone software application. In the latter case, the web browser

software would contain instructions for accessing the execution engine. Processor 11, memory 12, storage system 13, and display system 14 are coupled to bus 15, which preferably provides a high-speed means for  
5 devices connected to bus 15 to communicate with each other.

It will be apparent to one of ordinary skill in the art that computer system 10 is illustrative, and that alternative systems and architectures may be used  
10 with the present invention. It will further be understood that many other devices, such as a network interface (not shown), and a variety of other input and output devices (not shown) may be included in computer system 10.

Referring now to FIG. 2, an overview of a network environment in which a preferred embodiment of the present invention operates is shown. Client computer 16 ("client") uses web browser software to communicate across the Internet with server computer 17  
15 ("server"). Server 17 executes web server software to process requests from client computers on the Internet, such as client computer 16. Server 17 responds to these requests by sending electronic documents, executable code, or data to client 16, which may be  
20 part of an online computer language tutorial for display with the web browser software in client 16. In accordance with the present invention, interactive examples embedded in the tutorial are executed by an example execution engine on client 16.

30 Additionally, server 17 may contain "search engine" software for locating electronic documents on the Internet. Server 17 may also handle database management tasks, as well as a variety of administrative tasks, such as compiling usage

statistics. Alternatively, some or all of these tasks may be performed by computers 18a-c, connected to server 17 through local area network 19. It will be understood by one skilled in the art that there are  
5 millions of web servers connected to the Internet, and millions of client computers running web browser software. If server 17 has sufficient capacity, at any given time it may be in communication with thousands of client computers.

10           Web servers 20a-c are computers that provide access to electronic documents that may be found through use of a search engine provided by server 17. When a user of client 16 selects an electronic document from the search engine results provided by server 17,  
15 client 16 may communicate across the Internet with one of web servers 20a-c.

          Further, one skilled in the art will understand that the present invention could also be used in other network settings. For example, rather  
20 than connecting through the Internet, the apparatus and methods of the present invention could be used on a local area network. In such a configuration, the clients and server would all be connected to the same local area network. The methods and apparatus of the  
25 present invention may also be used in configurations in which the clients are also servers, that is, the clients and servers are the same computer. This would reduce the communications latency between user requests on the client and response to the user requests on the  
30 server.

          Referring now to FIG. 3, an example markup used to include coding examples in an electronic document that is part of a computer language tutorial is described. A "markup" refers to a sequence of

characters or other symbols that are inserted at certain places in the source code file of an electronic document to indicate how the electronic document should look when it is printed or displayed, or to describe the electronic document's logical structure. The markup indicators are often called "tags", and may comprise a starting tag and a termination tag. Tags are usually followed by source code content that is executed and displayed in the electronic document. In case the markup indicator consists of a starting tag and a termination tag, source code content is included between the starting tag and the termination tag. For example, using the <UL> (starting tag for an unnumbered list), </UL> (termination tag for an unnumbered list), and <LI> (list item) tags in the HTML language to display a list of fruits would require the following piece of source code in the electronic document:

```
<UL>
  <LI> apples
  <LI> bananas
  <LI> grapefruit
</UL>
```

The web browser software would display the following output in the web browser window:

- apples
- bananas
- grapefruit

Markup tags can be inserted by the electronic document creator directly by typing the symbols in the source code file of the electronic document, by using an editor and selecting prepackaged markup symbols (to save keystrokes), or by using a more sophisticated editor that enables users to create the document as they want it to appear.

Example markup 21 comprises example markup tag 22, which is defined according to the



specifications of the language used to create the electronic document. Example markup tag 22 may be the string "<example>" or any other string defined in the language specification for including source code for example content 23 in the electronic document. For example, in an online tutorial for HTML, an example markup may be created and placed in a tutorial for showing the user how to create a list of fruits as described above. The example markup may consist of the following source code:

```

    <example>
        This is a list of fruits.
        <UL>
            <LI> apples
15         <LI> bananas
            <LI> grapefruit
        </UL>
    </example>
```

The source code content included in the example markup is executed at the client computer by an example execution engine.

Referring now to FIG. 4, a flow chart for executing example markup 21 is described. At step 25, the example execution engine processes the example source code to produce the example result. The example source code may contain simple instructions for displaying text such as the list of fruits example above, or it may generate more sophisticated applications, involving audio, imagery, video, and user interactivity. An interactive example window is created at step 26 to display the example source code, the example result, and control mechanisms to enable users to easily and quickly modify the example source code, analyze the example results, debug the example source code, and repeat the process using the same

interactive example window. The interactive example window is displayed in a web browser window at step 27.

Referring now to FIG. 5, a preferred embodiment of an interactive example window for use with the present invention is described. Interactive example window 29 is a two-dimensional viewing area in the web browser display that allows users to view the example source code and results, as well as modify and execute the example. Interactive example window 29 contains three sub-windows, or widgets: example result widget 30, interactive editing widget 31, and control affordances widget 32. Example result widget 30 is a viewing area for displaying the results of the execution of the example source code content, displayed in interactive editing widget 31.

Interactive editing widget 31 is a simple editor that allows a user to modify the example source code content in editing widget 31 and directly typing the desired source code modifications. Preferably, editing widget 31 supplies common editing features such as copying and pasting. Operations on interactive editing widget 31 such as executing the example source code and restoring the displayed example source code to its original form are performed when the user interacts with control buttons provided in control affordances widget 32.

Referring now to FIG. 6, a preferred embodiment of a control affordances widget for an interactive example window is described. Control affordances widget 32 may include the following control buttons for accepting user interaction: "execute" control button 33, "revert" control button 34, "full edit" control button 35, "save as" control button 36, and "close pop-up" control button 37. When the user

selects execute control button 33, the example execution engine processes the example source code displayed in interactive editing widget 31 and displays the example results to the user. The example results  
5 may be displayed in example result widget 30, or in a "pop-up" window. Revert control button 34 may be selected by the user to display the original example source code in interactive editing widget 31. This option may be selected in case the user wants to  
10 compare the original example source code with any source code modifications the user makes in interactive editing widget 31.

A more complete editing window may be opened by the user by selecting full edit control button 35.  
15 Selecting this button causes a pop-up editor window to be displayed in the user's computer display screen. The pop-up window contains the example source code content displayed in interactive editing widget 31 and may be integrated with the development environment for  
20 the language used to write the example source code. This enables the user to more easily edit, execute, and debug the example source code. In case the user wants to save the example source code in a separate file, the user may select "save as" control button 36. "Close  
25 pop-up" control button 37 may be selected to close a pop-up window displaying example results.

Referring now to FIG. 7, an error message window for use with the present invention is described. Error message window 38 is a learning aid offered with  
30 the interactive example so the user may easily debug and understand mistakes made when modifying the example source code. Error message window 38 contains standard buttons 39, 40, and 41, for minimizing, maximizing, and closing the window, respectively. Error message 42

contains text revealing to the user the mistakes made when modifying the example source code, with possible suggestions as to where in the source code each mistake was made. Error message window 38 contains "view  
5 error line" control button 43 that, when selected, opens a full editor window displaying the modified source code. The line in the source code containing the error is highlighted in color or otherwise marked to call attention to the error. The user may then  
10 correct the error in the opened full editor window, copy the corrected source code text into interactive editing window 31, and execute the example by selecting execute control button 33.

Referring now to FIGS. 8A, 8B, 8C, 8D, and  
15 8E, an example of using the method of the present invention is described. Referring to FIG. 8A, interactive example window 45 shows exemplary CURL language source code for printing out a list of ancestors. The CURL content programming language and  
20 its features are described in the commonly owned, copending, U.S. patent application No. \_\_\_\_\_ (CURL-002).

Interactive example window 45 is displayed in an electronic document that is part of an online  
25 tutorial for the CURL language. The example source code is displayed in interactive editing widget 47, and the result of executing the example source code is shown in example result widget 46. The user may interact with the example by modifying the example  
30 source code displayed in interactive editing widget 47 and by using one of the control buttons displayed in control affordances widget 48.

Referring to FIG. 8B, interactive example window 45 shows that the example source code in

interactive editing widget 49 has been modified by changing "Your ancestors" to "My ancestors" and adding the ancestors' names. By clicking on the execute control button of control affordances widget 48, the  
5 modified example is executed by the example execution engine in the user's computer. Referring to FIG. 8C, example result pop-up window 50 is a pop-up window that displays the results of executing the modified example source code. Alternatively, the example results may be  
10 displayed in example result widget 46.

Referring now to FIG. 8D, interactive editing widget 51 shows modified example source code that contains typing error 52. When the user selects the execute control button in control affordances widget  
15 48, an error will occur. Referring now to FIG. 8E, error message window 53 is displayed to the user to indicate that an error occurred in the execution of the example source code. Error message window 53 shows error message 54, containing detailed information about  
20 the type of error made by the user. Error message window 53 also contains "view error line" control button 55 to open a full edit window displaying the example source code. The full edit window highlights the text line in the example source code that contains  
25 the error, thus facilitating the error debugging and the learning of the programming language.

Referring now to FIG. 9A, an example web browser window displaying an online coding tutorial containing an interactive example window is shown. Web  
30 browser window 56 displays text 57 and 59 explaining features of the CURL language. Example window 58 displays an example for computing the square root of a number and can be modified by a user as described above.

The example source code for displaying the online coding tutorial text and example shown in FIG. 9A in a web browser window is shown in FIG. 9B. The example source code in the CURL language contains

5 paragraph markup 60 and example markup 61 for displaying a text paragraph and an interactive example window, respectively. Text paragraph 62 refers to a feature of the CURL language used in example 63. Example markup 61 causes example 63 to be displayed in

10 interactive example window 58 of FIG. 9A.

Although particular embodiments of the present invention have been described above in detail, it will be understood that this description is merely for purposes of illustration. Specific features of the

15 invention are shown in some drawings and not in others, and this is for convenience only and any feature may be combined with another in accordance with the invention. Steps of the described processes may be reordered or combined, and other steps may be included. Further

20 variations will be apparent to one skilled in the art in light of this disclosure and are intended to fall within the scope of the appended claims.

Further, one skilled in the art will understand that the methods and apparatus of the

25 present invention could also be used to provide examples for a variety of online tutorials, such as tutorials for architectural software packages, tutorials for electrical circuit simulators, and tutorials for network simulators, among others.

What Is Claimed Is:

1. A method for generating an interactive coding example for an online computer language tutorial, the method comprising:
  - defining an example markup for including an example source code in the online tutorial;
  - processing the example source code to produce an example result;
  - creating an interactive example window; and
  - displaying the interactive example window in the online tutorial.
2. The method of claim 1, wherein defining an example markup for including an example source code in the online tutorial comprises defining an example markup in the language used to create the online tutorial.
3. The method of claim 1, wherein processing the example source code in the online tutorial comprises processing the example source code by an example execution engine.
4. The method of claim 3, wherein the example execution engine comprises instructions for executing source code examples in online computer language tutorials.
5. The method of claim 1, wherein creating an interactive example window comprises
  - creating an interactive editing widget;
  - creating an example result widget; and
  - creating a control affordances widget.

6. The method of claim 5, wherein the interactive editing widget comprises an editor window for typing example source code.

7. The method of claim 5, wherein the example result widget comprises a window for displaying example results.

8. The method of claim 5, wherein the control affordances widget comprises user interaction control buttons.

9. The method of claim 8, wherein the control buttons comprise one or more control buttons, selected from a group consisting of:

- an execute control button;
- a revert control button;
- a full edit control button;
- a save as control button; and
- a close pop-up control button.

10. The method of claim 9, wherein the execute control button comprises processing the example source code and displaying the example results to the user when the execute control button is selected by the user.

11. The method of claim 10, wherein the execute control button further comprises generating an error message window when the execute control button is selected and there is an error in the example source code.



12. The method of claim 11, wherein generating the error message window comprises displaying an error message indicating the error in the example source code.

13. The method of claim 9, wherein the revert control button comprises displaying the original example source code to the user when the revert control button is selected by the user.

14. The method of claim 9, wherein the full edit control button comprises displaying a pop-up editor window for editing the example source code when the full edit control button is selected by the user.

15. The method of claim 9, wherein the save as control button comprises saving the example source code in a file when the save as control button is selected by the user.

16. The method of claim 9, wherein the close pop-up control button comprises closing the pop-up window for displaying example results when the pop-up control button is selected by the user.

17. An apparatus for generating an interactive coding example for an online computer language tutorial, the apparatus comprising:  
a processor;  
a display system; and  
a memory, the memory storing a plurality of routines for execution by the processor, wherein the plurality of routines comprise:  
web browser software routines; and

an example execution engine;

18. The apparatus of claim 17, wherein the example execution engine comprises routines for executing source code examples in online computer language tutorials displayed in a web browser window.

19. The apparatus of claim 18, wherein the routines for executing source code examples comprise:  
routines that are embedded in web browser software;  
routines that are a plug-in to web browser software; and  
stand-alone routines that are accessed by web browser software.

Abstract of the Disclosure

Methods and apparatus for generating an interactive coding example for an online computer language tutorial are shown. The interactive coding  
5 example is displayed in an electronic document as part of an online computer language tutorial. An interactive example window is displayed in the electronic document by web browser software to enable the user to easily view, modify, execute, and debug the  
10 example instructions. The example source code is processed by an example execution engine in the user's computer.

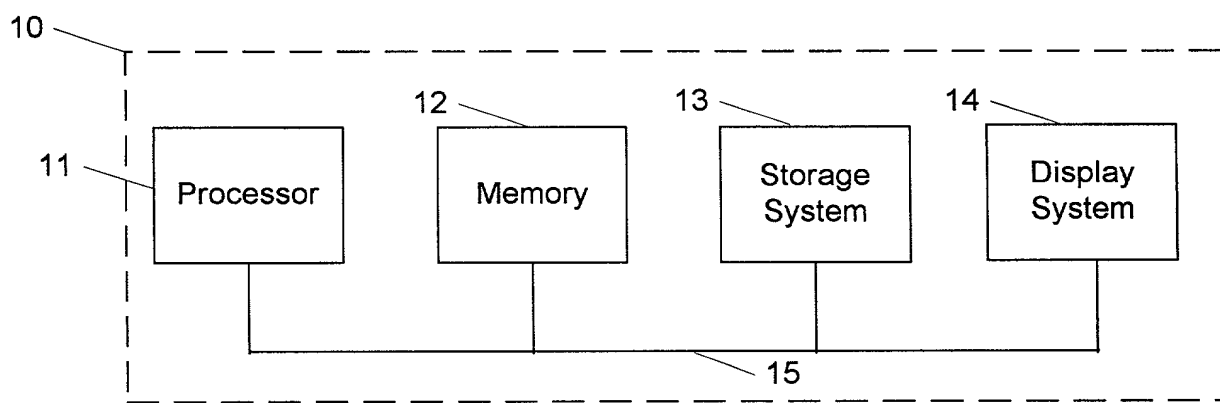


FIG. 1

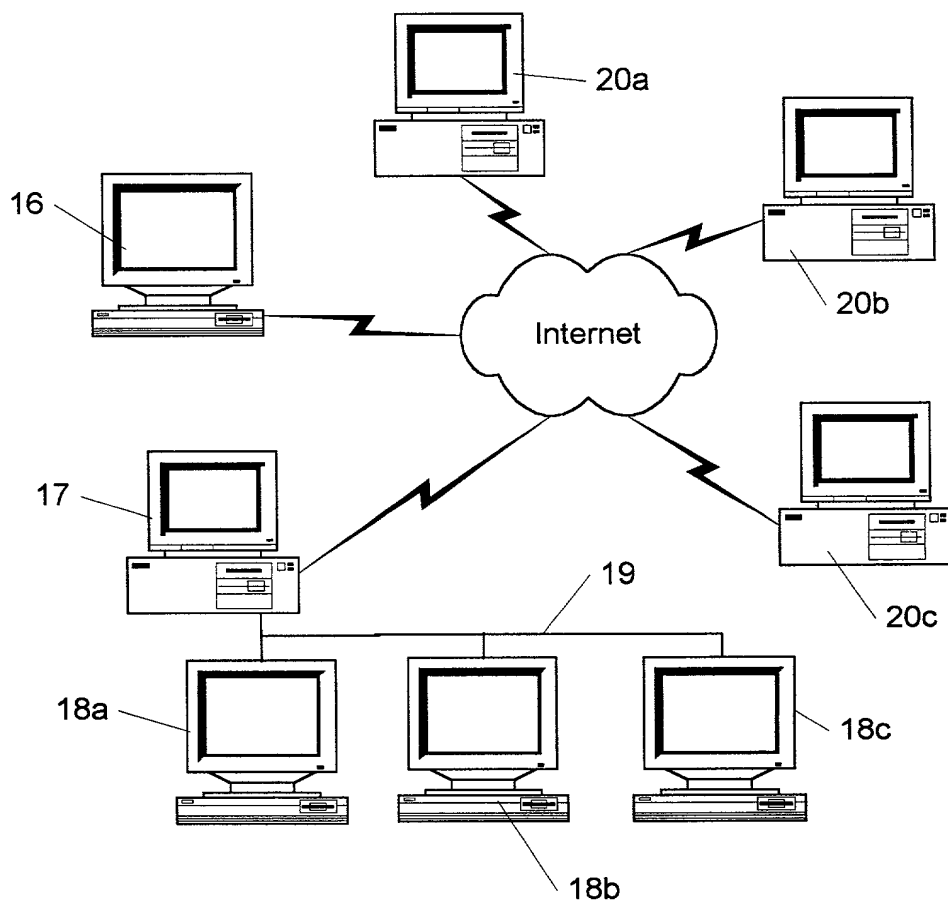


FIG. 2

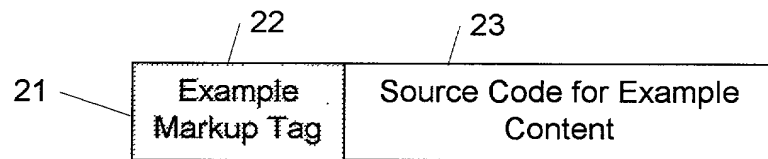


FIG. 3

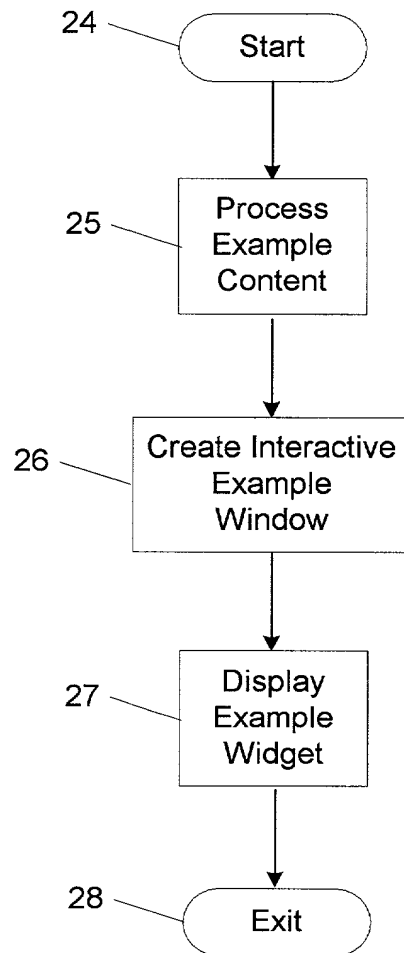


FIG. 4

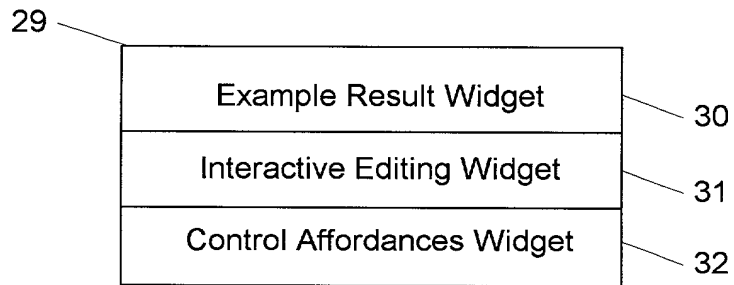


FIG. 5

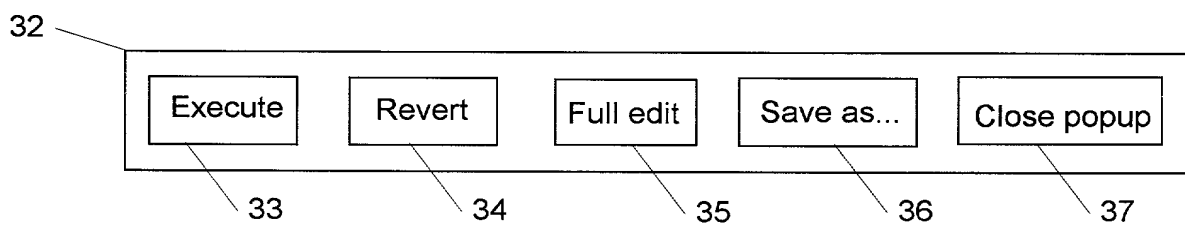


FIG. 6

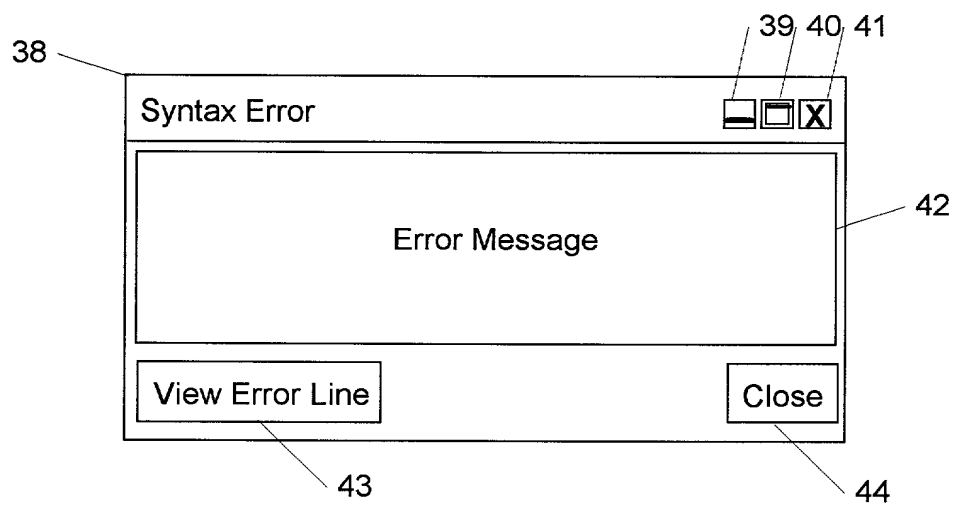


FIG. 7

**Ancestors:** Your ancestors include

2 *parents*  
4 *grandparents*

|| Print out number of ancestors

{**Ancestors:**} Your ancestors include

{Table columns=2,  
2, {*parents*},  
4, {*grandparents*}  
}

Execute

Revert

Full edit

Save as...

Close popup

FIG. 8A

**Ancestors:** Your ancestors include

2 *parents*  
4 *grandparents*

|| Print out number of ancestors

{**Ancestors:**} My ancestors include

{Table columns=2,  
2, {*parents: Charles and Ann*},  
4, {*grandparents: Roger, JoAnn, Thomas, and Gisele*}  
}

Execute

Revert

Full edit

Save as...

Close popup

FIG. 8B

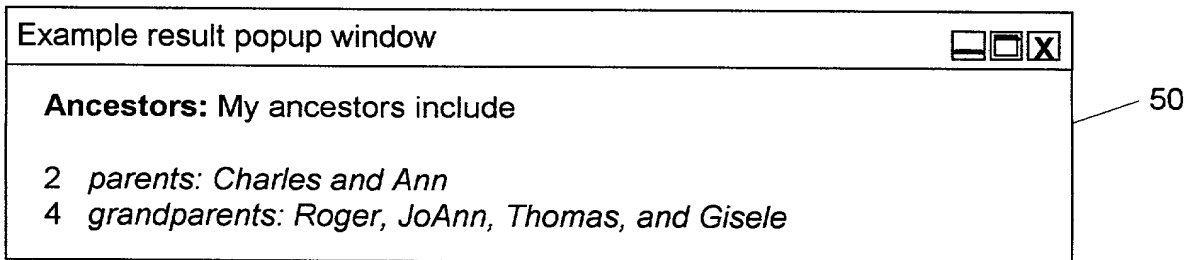


FIG. 8C

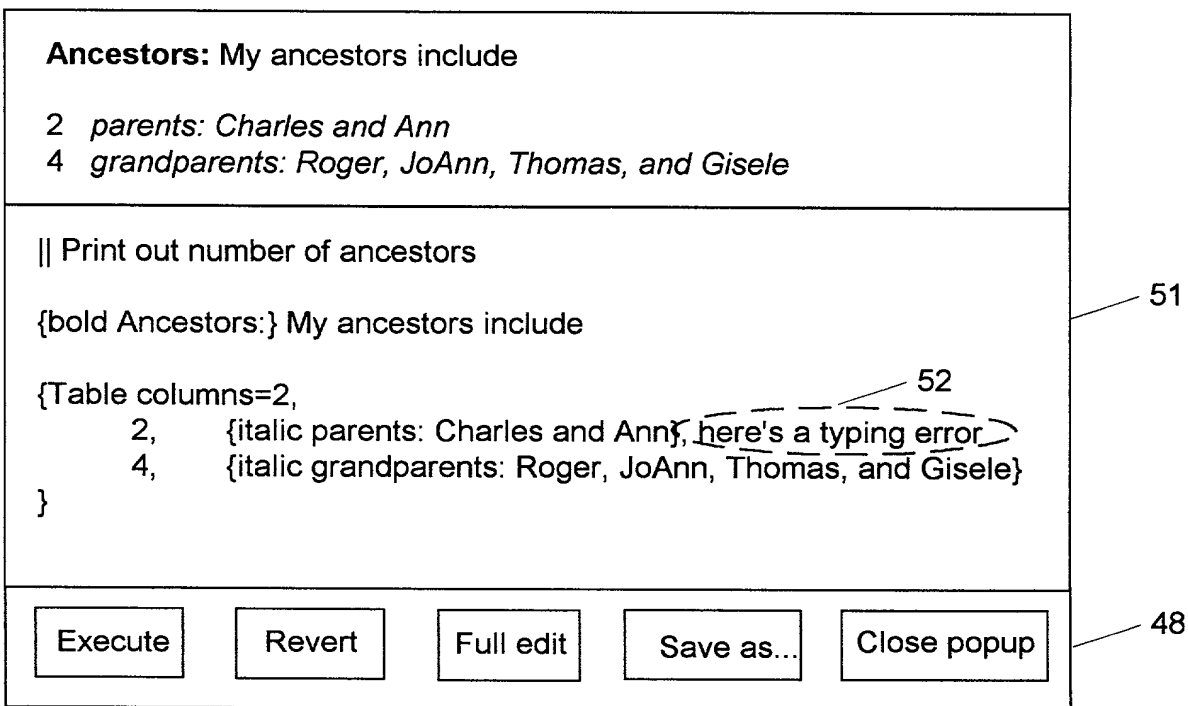


FIG. 8D

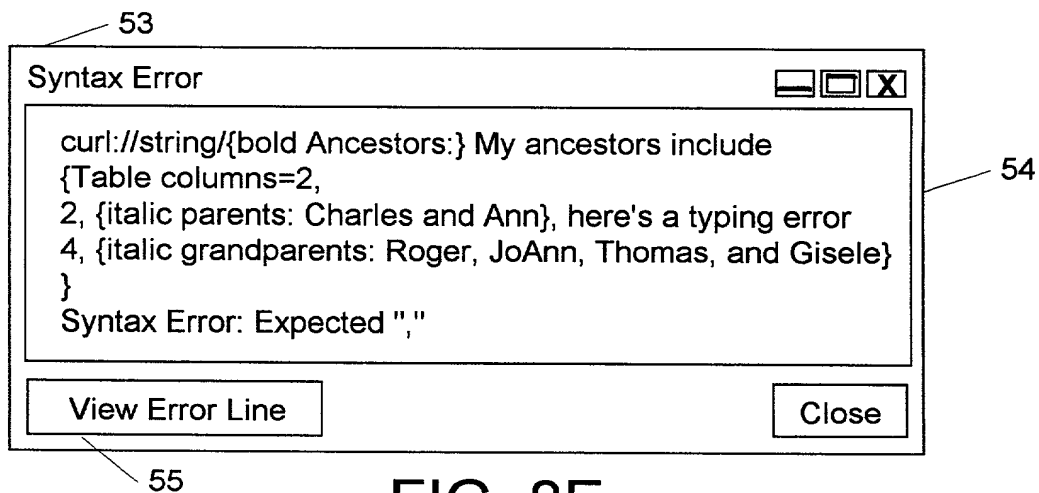


FIG. 8E



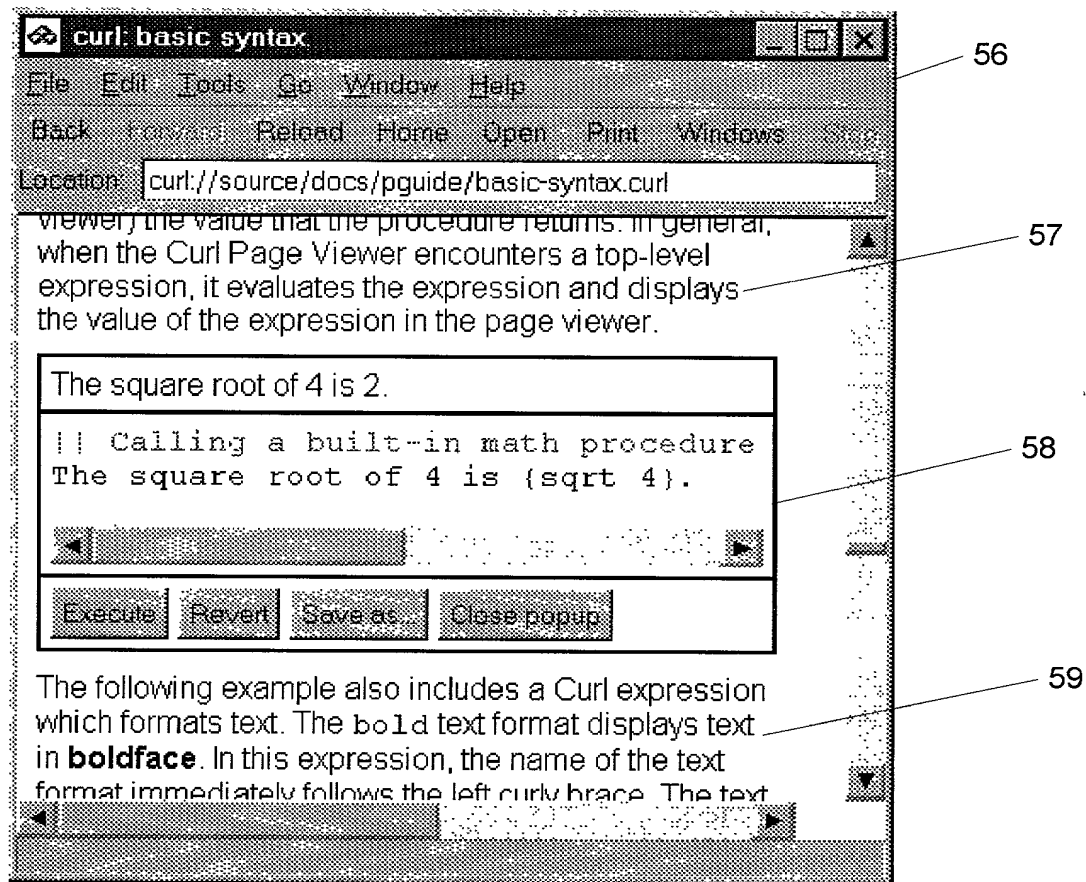


FIG. 9A

{paragraph 60

The following interactive example includes a comment, some text, and a Curl expression. The Curl expression in this example is a call to the {code sqrt} procedure, which is built into Curl. (Like many programming languages, Curl includes many convenient features like the {code sqrt} procedure that are built into the language. The {code sqrt} procedure takes a value that you supply and returns the square root of that value.) In an expression, the name of the procedure must immediately follow the left curly brace. The value or values that you supply to the procedure follow the name of the procedure, and the right curly brace indicates the end of the expression. When Curl encounters the expression in this example, it calls the {code sqrt} procedure, passing the value supplied to the procedure. It then displays (in the page viewer) the value that the procedure returns. In general, when the Curl Page Viewer encounters a top-level expression, it evaluates the expression and displays the value of the expression in the page viewer.

62

}  
{example 61

|| Calling a built-in math procedure 63  
The square root of 4 is {sqrt 4}.

}

FIG. 9B

CURL-006

**DECLARATION AND POWER OF ATTORNEY  
FOR PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

INTERACTIVE EXAMPLES FOR ONLINE CODING TUTORIALS

the specification of which

(check ☒ [X] is attached hereto  
one)

☐ [] was filed on \_\_\_\_\_ as  
Application Serial No. \_\_\_\_\_ and  
was amended on \_\_\_\_\_.  
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I do not know and do not believe that the invention was ever patented or described in any printed publication in any country before my or our invention thereof or more than one year prior to this application.

I do not know and do not believe that the invention was in public use or on sale in the United States of America more than one year prior to this application.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known by me to be material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or

inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

			<u>Priority Claimed</u>	
			<input type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
			<input type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known by me to be material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

As a named inventor, I hereby appoint the following attorneys or agents to prosecute this application and transact all business in the United States Patent and Trademark Office connected therewith:

Nicola A. Pisano, Esq. (Reg. No. 34,408)
Michael J. DeHaemer, Jr., Esq. (Reg. No. 39,164)
Christopher J. Frerking (Reg. No. 42,557)

Send correspondence to:

Nicola A. Pisano  
FISH & NEAVE  
1251 Avenue of the Americas  
New York, New York 10020-1104

Direct telephone calls to:

Nicola A. Pisano  
(650) 617-4018

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of first joint inventor Robert H. Halstead

First Inventor's signature \_\_\_\_\_ Date \_\_\_\_\_

Residence 24 Louise Road, Belmont, MA 02178

Citizenship United States of America

Post Office Address 24 Louise Road

Belmont, MA 02178

Full name of second joint inventor Patrick J. LoPresti

Second Inventor's signature \_\_\_\_\_ Date \_\_\_\_\_

Residence 61A 6th Street, Cambridge, MA 02141

Citizenship United States of America

Post Office Address 61A 6th Street

Cambridge, MA 02141